



Univerzita
Pardubice
Fakulta elektrotechniky
a informatiky

UNIVERZITA PARDUBICE

FAKULTA ELEKTROTECHNIKY A INFORMATIKY

KATEDRA SOFTWAREVÝCH TECHNOLOGIÍ

OBJEKTIVĚ ORIENTOVANÉ PROGRAMOVÁNÍ - IOOP

ZADÁNÍ SEMESTRÁLNÍ PRÁCE

Evidence dopravních prostředků

Autor zadání:
Karel ŠIMERDA

Garant předmětu:
Karel ŠIMERDA

Verze 3.1
2.11.2020

1 Úvod

Cíl semestrální práce Seznámit studenty se základy objektově orientovaného programování na malém projektu.

Účel dokumentu Tento dokument obsahuje zadání a postup prací na semestrální práci „Evidence dopravních prostředků“.

Zadání semestrální práce navazuje na úlohy, které byly řešeny v rámci předmětu IPALP a předchozí úloze předmětu IOOP U04. Většina studentů je tedy seznámena s problémovou doménou evidence automobilu a nyní se můžou více soustředit na pracovní postupy objektového přístupu a na implementaci samotného zdrojového kódu.

Úprava původní verze evidence automobilu z předmětu IPALP bude spočívala v tom, že se nejdříve vytvoří třída se spojovým seznamem, která bude nezávislá na typu vkládaných datových prvků. Tato třída může procházet postupně změnami, na kterých bude ukázán objektový přístup a ověření jednotkovým testem. Při realizaci grafického rozhraní a jeho propojení se seznamem dopravních prostředků budou představeny další možnosti syntaxe jazyka, jako je například genericita, rozšířená syntaxe v interfejsu ve smyslu Java 8, lambda výrazy a tzv datovody (streams).

Teprve po dokončení vývoje obecného rozhraní a třídy spojového seznamu, bude přikročeno k jejich vytvoření tříd různých typů dopravních prostředků.

Při vývoji budeme používat princip výstavby programu: Ze zdola nahoru. To znamená, že nejdříve naprogramujeme třídy nejnižší úrovně, které nepoužívají jiné třídy. Teprve později přikročíme k naprogramování tříd vyšší úrovně a až naposled vytvořené třídy začleníme do uživatelského rozhraní.

2 Vize

Program „**Evidence dopravních prostředků**“ bude umožňovat vést seznam různých typů prostředků, jako jsou osobní automobily, dodávky, nákladní automobily nebo traktory. Tento omezený počet typů dopravních prostředků naprosto stačí pro naše výukové potřeby. Jde o princip, jak pracovat s několika různými typy datových entit. V reálné aplikaci by samozřejmě bylo typů daleko víc a jednotlivé typy by obsahovaly více informací.

Nejdříve si naprogramujeme uživatelské rozhraní v příkazovém řádku (CL) a potom grafické (GUI).

Řízení programu v GUI bude zajišťovat několik ovládacích prvků v hlavním okně aplikace. Tyto prvky budou umožňovat spouštění funkcí zobrazování, setřídění, vyhledávání nebo filtrování uložených informací o dopravních prostředcích.

Z hlavního okna se budou vyvolávat jednotlivá dialogová okna, v kterých se budou zadávat nebo měnit parametry dopravních prostředků.

Bude požadováno, aby stav programu bylo možné trvale uložit do souboru a zpětně obnovit a to jak pomocí serializace, tak i textové formě.

V programu bude vyžadováno co možná nejširší použití vlastností Java 8. Rozsah využití syntaxe Java 8 bude mít vliv na závěrečné hodnocení studenta.

3 Požadavky

3.1 Funkční požadavky

3.1.1 Požadavky na datovou entitu

FR_01 Typy dopravních prostředků: Požaduje se, aby aplikace vedla nejméně čtyři typy dopravních prostředků a to

- osobní automobil
- dodávka
- nákladní automobil
- traktor

FR_02 Charakteristika typů dopravních prostředků: Požaduje se, aby každý typ byl popsán alespoň dvěma společným údaji (SPZ a hmotnost) pro všechny typy a alespoň jedním jedinečným údajem. Alespoň u jednoho typu bude použit vlastní výčet barev.

FR_03 Rozdílné charakteristiky typů dopravních prostředků: Každý student si vytvoří vlastní jedinečnou charakteristiku (seznam atributů) u každého typu v jejich počtech, názvech a významech. Shoda v charakteristice typů u dvou nebo více projektů, bude považována za nesplnění zadání semestrální práce.

3.1.2 Požadavky na příkazový řádek (CL)

FR_04 Příkazy CL Program bude v režimu příkazového řádku mít příkazy, které jsou uvedeny v příloze 8.1 Příkazy CommandLine.

FR_05 Ukončení programu Program bude v režimu příkazového řádku příkazy vykonávat v cyklu do té doby než bude uživatelem zvolen příkaz exit.

3.1.3 Požadavky na grafické uživatelské rozhraní (GUI)

FR_10 Zobrazování seznamu: Program bude v hlavním okně zobrazovat seznam dopravních prostředků včetně charakteristik každého typu.

FR_11 Generuj obsah seznamu: Stisknutím tlačítka **Generuj** se nejdříve zjistí počet kolik se jich má vytvořit a poté se nahraje do seznamu zkušební seznam dopravních prostředků sestavený generátorem dopravních prostředků a seznam se zobrazí.

FR_12 Nový prostředek: Tlačítkem nebo ComboBox **Nový** se otevře dialogové okno se zadáním nového prostředku. Po zvolení typu prostředku se zobrazí příslušné parametry prostředku, aby bylo možné zadat hodnoty. Po vložení nové nebo změněné položky se obnoví zobrazení seznamu. Každý student si zvolí vhodné parametry pro každý typ prostředku podle svého uvážení. Požaduje se nejméně 3 parametry, přičemž alespoň jeden musí být společný pro všechny typy.

FR_13 Výběr položky: V poli se zobrazeným seznamem prostředků bude možné vybrat myší jednu položku k dalším operacím.

FR_14 Zrušení prostředku v seznamu: Tlačítkem **Vyjmi** se zruší vybraný prostředek ze seznamu a seznam se obnoví.

FR_15 Editace parametru dopravního prostředku: Tlačítkem **Edituj** se otevře dialogové okno vybraného prostředku podle jeho typu. V dialogovém okně se zobrazí příslušné parametry prostředku k změně hodnot. Po ukončení změn v dialogovém okně se seznam prostředků automaticky obnoví.

FR_16 Filtrace podle typu dopravních prostředků: Po výběru typu dopravních prostředků v ComboBox se v hlavním okně zobrazí pouze prostředky požadovaného typu.

FR_17 Zrušení filtrace: Stisknutím tlačítka **Zruš filtraci** dojde ke zrušení filtrace nebo volbou v ComboBox **nefiltruj**.

FR_18 Uložení zálohy: Stisknutím tlačítka **Zálohuj** dojde k uložení všech prostředků v seznamu do záložního binárního souboru. Do souboru se budou ukládat pouze serializované objekty prostředků. Jméno souboru bude pevné a to **zaloha.bin**.

FR_19 Obnovení seznamu: Stisknutím tlačítka **Obnov** dojde k nahrazení aktuálního seznamu seznamem prostředků ze souboru **zaloha.bin**.

FR_20 Uložení seznamu prostředků do textového souboru: Stisknutím tlačítka **Ulož** dojde k uložení všech prostředků v seznamu do textového souboru. Jméno souboru bude pevné a to **prostredky.txt**.

FR_21 Obnovení seznamu z textového souboru: Stisknutím tlačítka **Načti** dojde k nahrazení aktuálního seznamu seznamem prostředků ze souboru **prostredky.txt**.

3.2 Nefunkční požadavky

3.2.1 Požadavky na implementaci třídy Seznam

NRS_01 Spojový seznam: Požaduje se implementovat třídu **LinSeznam** jako jednosměrný spojový seznam, který bude nezávislý na typu vkládaných dat. Pro tento seznam nesmí být použita žádná třída z knihovny Java. Tato třída musí být zcela plně nově naprogramována. Třidu ověřte testem JUnit.

NRS_04 Rozhraní Seznam: Požaduje se, aby obě třídy **SpojovySeznam** a **SeznamNaPolí** implementovaly rozhraní **Seznam**, které je součástí tohoto zadání. Je zakázáno toto rozhraní měnit, pokud nebude v něm zjištěna chyba.

NRS_05 Omezení veřejných metod implementační třídy: Třída **LinSeznam** nesmí mít další veřejné metody než ty, které jsou dány rozhraním **Seznam**.

NRS_06 Serializace objektů: Požaduje se, aby ukládání obsahu seznamu pomocí serializace do souborů a jejich opětovné obnovení a **SeznamNaPolí**. Na to je určena třída, která bude implementovat interfejs **Ovladani**.

NRS_07 Procházení seznamu: Požaduje se, aby prohlídka seznamu byla zajištěna pomocí vnitřní nebo anonymní třídy, která bude implementovat rozhraní **Iterator** z knihovny Java.

3.2.2 Požadavky na implementaci uživatelského rozhraní v příkazovém řádku

NFRCL_01 Rozhraní v příkazovém řádku: Požaduje se, aby funkce interfejsu **Ovladani** bylo možné volat z příkazového řádku.

3.2.3 Požadavky na implementaci grafického uživatelského rozhraní

NFRGUI_01 Rozhraní JavaFX: Požaduje se, aby grafické uživatelské rozhraní (GUI) bylo realizováno v JavaFX.

NFRGUI_02 Tvorba grafického rozhraní: Požaduje se, aby sestavení scén (scene) na jevišti (stage) byly vytvářeny pouze prostředky jazyka Java. Není tedy povoleno použití programu Scene-Builder pro tvorbu scén.

3.2.4 Požadavky na architekturu aplikace

NRAPP_01 Ošetření chyb: Požaduje se, aby chyby ve vykonávání metod tříd byly hlášeny pouze pomocí výjimek a obslouženy třídou **Alert**.

NRAPP_02 Použití datových proudů Stream: Požaduje se, aby aplikace využívala k výběru obsahu seznamu přednostně datových proudů (datovodů).

NRAPP_03 Použití lambda-výrazů: Požaduje se, aby pro filtraci položek při výběru ze seznamu byly použity lambda výrazy.

NRAPP_04 Používání výčtů: Požaduje se, aby hodnoty výčtů byly zobrazovány přirozeným textem. Přičemž tento text hodnot nesmí být součástí GUI.

NRAPP_05 Oddělení výkonného kódu od GUI: Požaduje se, důsledné oddělení grafického uživatelského rozhraní od výkonného kódu aplikace. Doporučuje se použít nějaký vhodný návrhový vzor, jako jsou například vzory adaptér nebo fasáda. To bude zajišťovat implementace rozhraní **Ovladani**.

3.2.5 Požadavky na vedení projektu

NRP_01 Ukládání projektu Požaduje se projekt ukládat do verzovacího systému Subversion (SVN) na adrese <https://fei-svn.upceucebny.cz> do složky **ioop** studenta.

NRP_02 Omezení souborů v úložišti SVN Požaduje se ukládat do úložiště SVN pouze zdrojové a projektové soubory. Je zakázáno ukládat soubory, které byly získány překladem nebo generátorem Javadoc.

NRP_03 Verzování projektu Je povoleno mít na úložišti ve složce **ioop** jen jeden projekt aplikace, který bude postupně verzován. V případě potřeby si může student vytvořit vývojovou větev projektu, ale mimo složku **ioop**. Vývojová větev projektu se musí vytvořit v rámci SVN a ne tím, že se nainportuje jako nový projekt!

4 Postup prací

1. Založit projekt v Netbeans s názvem **EvidenceProstredku_prijmeni**, jehož součástí bude příjmení studenta.
2. Přednastavit tyto povinné balíčky

kolekce pro všechny třídy, a rozhraní, které souvisí s interfejsem **Seznam**

generator pro třídy generátoru testovacího vzorku dopravních prostředků.

perzistence pro třídy, které zajistí lehkou perzistenci kolekce dopravních prostředků do/ze souborů.

prostredky pro třídy, které popisují typy dopravních prostředků

sprava pro rozhraní `Ovladani` a jeho implementační třídy

command pro třídy uživatelského rozhraní příkazového řádku

gui pro třídy grafického uživatelského rozhraní

3. Do balíčku kolekce vložit rozhraní `Seznam`, a vytvořit třídu `LinSeznam` a další potřebné soubory.
4. Třídy `LinSeznam` ověřit testovací třídou v JUnit.
5. Teprve po vývoji a otestování třídy `LinSeznam` pokračovat ve vývoji ostatních tříd, rozhraní a výčtů v ostatních balíčcích.
6. Pro ostatní třídy se nepožaduje otestování testy JUnit. Tyto třídy budou ověřeny pouze přes uživatelské rozhraní aplikace.

5 Harmonogram

Studentům se doporučuje zachovat následující harmonogram, protože v souladu s ním budou probírány jednotlivé části semestrální práce na přednáškách a na cvičeních zase bude probíhat inspekce kódu.

do 6.11. Dokončení třídy `LinSeznam`.

do 13.11 Jednoduché ovládání třídy `LinSeznam` z příkazového řádku.

do 20.11. Implementace rozhraní `Ovladani` podle návrhového vzoru fasáda.

do 27.11. Implementace uživatelské rozhraní v příkazovém řádku přes rozhraní `Ovladani`

do 11.12. Dokončení semestrální práce implementací grafického uživatelského rozhraní (GUI). Zároveň je to mezní termín odevzdání semestrální práce.

6 Hodnocení práce a její využití u zkoušky

1. Pro získání zápočtu je nutné odevzdat semestrální práci se dvěma uživatelskými rozhraními.
2. Na praktické části zkoušky může být požadováno doplnit projekt semestrální práce o další funkčnost nebo bude požadováno nějakou část kódu přeprogramovat.
3. Na ústní části zkoušky může být požadováno, aby student vysvětlil nějakou část zdrojového kódu této semestrální práce.
4. Na zkoušce bude hodnoceno do jaké míry student splnil v projektu semestrální práce zde uvedené požadavky. Dále bude hodnoceno, jak student respektoval doporučení, jak postupovat z přednášek a ze cvičení.

7 Poznámky a upozornění

Poznámka: Po vytvoření projektu a vložení rozhraní **Seznam**, lze vygenerovat dokumentaci pomocí Javadoc v menu Run v NetBeans. Po vygenerování dokumentace lze kontrakt a signaturu rozhraní a i dalších tříd nebo výčtů projektu prohlížet internetovém prohlížeči. Vygenerované soubory html neukládat do úložiště SVN.

Upozornění: Případné nesrovnalosti, které se můžou vyskytnout v tomto zadání, budou upřesněny učitelem na cvičení. V případě závažných připomínek bude vydána další verze tohoto dokumentu.

8 Přílohy

8.1 Příkazy CommandLine

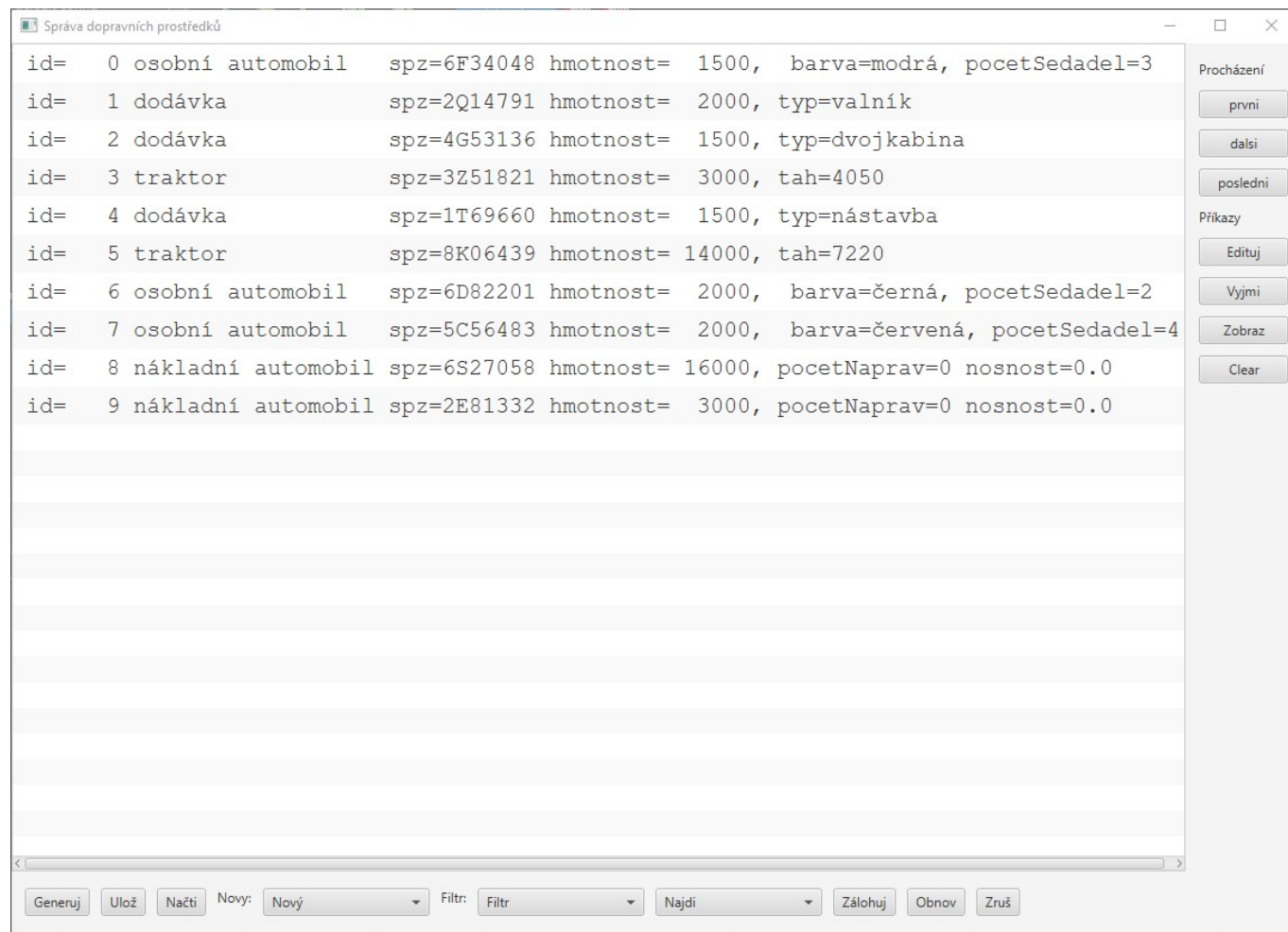
help, h	- výpis příkazů
novy,no	- vytvoř novou instanci a vlož dopravního prostředku za aktuální
najdi,na,n	- najdi v seznamu dopravní prostředek podle státní poznávací značky
dej	- zobraz aktuální dopravní prostředek v seznamu
edituj,edit	- edituj aktuální dopravní prostředek v seznamu
vyjmi	- vyjmi aktuální dopravní prostředek ze seznamu
prvni,pr	- nastav jako aktuální první dopravní prostředek v seznamu
dalsi,da	- přejdi na další dopravní prostředek
posledni,po	- přejdi na poslední dopravní prostředek
pocet	- zobraz počet položek v seznamu
obnov	- obnov seznam dopravních prostředků z binárního souboru
zalohuj	- zalohuj seznam dopravních prostředků do binárního souboru
vypis	- zobraz seznam dopravních prostředků
nactitext,nt	- načti seznam dopravních prostředků z textového souboru
uloztext,ut	- ulož seznam dopravních prostředků do textového souboru
generuj,g	- generuj náhodně dopravní prostředky pro testování
zrus	- zruš všechny dopravní prostředky v seznamu
exit	- ukončení programu

8.2 Formát textového souboru pro načtení nebo uložení seznamu

```
na, AAAAAA, 4500, 0,000000
na, 8006113, 1000, 0,000000
tr, 7B68580, 12500, 8710
oa, 5F96291, 1500, 8, bílá
tr, 4U48741, 4000, 4670
na, 7F05929, 12500, 0,000000
oa, 9Q95684, 2000, 4, červená
tr, 6P06862, 8000, 3630
do, 8X35824, 1000, dvojkabina
tr, 8Z07272, 8000, 5270
do, 1[15374, 2500, nástavba
do, 9V81293, 3000, dvojkabina
tr, 1M93794, 12500, 1210
na, 7U67962, 1500, 0,000000
do, 3D96324, 1000, dvojkabina
na, 8D49789, 17000, 0,000000
na, 5N52147, 15000, 0,000000
```

na, 8Y86975, 7500, 0,000000
na, 1D27432, 11000, 0,000000
do, 2V55132, 2500, dvojkabina

8.3 Hlavní okno aplikace (GUI)



Obrázek 1: Okno aplikace